一套針對 Windows 系統架構之木馬程式偵測排除機制

論文摘要

張簡維仁
中華電信研究所
williamc@cht.com.tw

摘要

隨著駭客技術的演進，現行資訊安全防護體系對於木馬程式的入侵及隱藏已經越來越難以防範。鑒於當前資訊系統內部隨時可能出現木馬藏匿的情況，建立一套可行且有效的木馬偵測清除方式已是當務之急。在本文中，藉由分析木馬程式行為並結合木馬防治的實務經驗，我們據此架構出一套偵測及清除 Windows 系統中木馬程式的機制。透過在實際環境下的操作，可以確認此機制具有相當的可靠性。同時我們也希望藉由此文，提供系統管理者作為木馬防治的參考。

關鍵詞：木馬偵測，木馬行為

一套針對 Windows 架構之木馬程式偵測排除機制

壹、緒論

在資訊安全領域中，傳統上對於木馬、病毒等惡意程式的防治通常採取”防火牆-入侵偵測系統-防毒軟體”的架構來進行。然而隨著駭客技術的演進，其入侵系統的方式日益細膩且難以察覺。尤其是諸如社交攻擊與網頁釣魚等攻擊方式，更是使得防火牆與入侵偵測等傳統資安系統對於木馬等外來惡意程式的防堵益形困難。而此時同樣發展快速的木馬匿蹤技術，則讓防毒軟體在查測清除現存系統中新型木馬的能力備受打擊。除此之外，根據資料統計，在美國 85%以上的企業與政府單位往往存在安全性漏洞（Kleiman 2007）。這些系統漏洞尤其提供了木馬程式極佳的入侵與隱藏的環境。事實上，現今大部分系統並不存在內部有無木馬存在的問題，而是隱藏的木馬何時才能夠被發現的問題。然而儘管如此，木馬偵測技術的研究在目前似乎卻仍未受相應的重視。有鑑於此，本文嘗試將筆者工作實務上實際偵測清除 Windows 系統上木馬的方式加以整理並系統化，以提供有志於此領域的研究者作一參考。

截至目前為止，對於系統中木馬程式的偵測技術大致可分為三種。第一種類型的偵測方是主要是透過比對木馬程式內含的特徵碼（即俗稱的病毒碼）來查測系統中是否存在木馬的隱藏。此種掃描特徵碼的偵測方式對於已被記錄確認的木馬雖然辨識率頗高，但相對地對於未知病毒碼的木馬卻無法有效防範。有鑑於目前新種木馬出現的速度遠快過特徵碼被確認的速度，此種偵測方式無法完全防範隱藏於系統中的木馬程式。不過因為此種方式簡單易用，因此廣為目前大多數防毒軟體所採用。

第二種木馬偵測方式主要在偵測系統中是否存在木馬程式設置的系統叫用掛鉤（system call hook）（梁曉 等 2007）。由於時下新一代木馬大多透過攔截 Windows API 的方式達成其於系統中匿蹤之目的（林小進、錢江 2007），因此藉由分析系統使用者程序與核心程序中關於系統函數叫用的標記訊息，即可查知系統中是否存在木馬程式所設置的隱藏式系統叫用掛鉤，連帶得以判定系統中是否存在木馬。此種方式雖能發現藏匿於系統中未知的木馬，然而一旦木馬採取直接修改系統核心的方式，則此方式將相對地難以有效進行隱藏木馬的偵測。

第三種木馬偵測方式統稱為行為偵測模式（Jie Qin et. al. 2010）。此種偵測模式主要根據分析木馬的各種行為模式來架構偵測方法，以避免產生前兩種方式無法找特定類型木馬的盲點。目前此等模式包含了下列幾種研究中的方法：資料探勘方式（Shugang Tang 2009）與程序追蹤方式（Naiqi Wu et. al. 2006）。基本上來說，此等偵查方式的發展，目前仍較偏向理論方面的研究。

由上述木馬程式偵測的技術觀之，本論文中所提出討論的木馬偵測方式原則上較偏向第三種，亦即根據木馬行為模式來架構其對應的偵測技術。與其他同種偵測模式的方法相較，本文提出的方式主要建立在實務經驗的判斷上，這是本文方法較為不同之處。

貳、偵測清除木馬機制

透過實務上的觀察我們發現，儘管不同類型的木馬實作方式不盡相同，其目的與行為模式卻大同小異。對於木馬程式而言，其在被植入的電腦系統中的動作不外乎隱藏自身、蒐集資訊並透過對外連線將資料傳出系統之外。

對木馬的設計、製作或散布者來說，如何獲取被木馬入侵系統中貴重的資料，理應是他們最為關切的目的。因此不論木馬的型態為何，在運作執行階段必然會與外界存在網路連線。基於此種特性，傳統上許多資安相關文章或文獻會建議將掃描系統通訊埠的連線狀況列為偵測木馬存在與否的關鍵步驟。然而隨著木馬匿蹤技術的精進，前述方式在實務上已經難以作為檢測系統木馬的主要方式。造就現今木馬難以偵測的結果主要根源於下列幾個因素：(1)端口反彈型木馬技術（阮寧君 2007）的出現，使得現今木馬多透過如 HTTP 等防火牆通常會開放的通訊埠與外界聯繫。(2)為了躲避偵測，現今木馬多刻意以定期或不定期的方式對外連線，以取代過去易被發現的持續性連線。(3)運用 system call hook 的技術，將自身執行程序隱藏於合法系統程序或服務之中（一般稱此為 rootkit 木馬），如此即使其通訊被偵測到，也會因難以判定而被忽略。而上述的第三點，除了混淆連線程式的判定外，也與木馬執行時期的匿蹤技術有相當關係。

除了具備與外界連線以傳遞資料訊息的特性外，如何巧妙的隱藏自身以蟄伏於系統中運行而不被察覺，也是設計者加諸於木馬程式上的關鍵技術。相較於早期獨立占據一個執行程序運行的前代木馬，現今實務上發現的木馬大多採用動態程式庫連接（DLL）的方式，將自身隱藏加載在系統合法程序或服務的背後來執行。此種匿蹤技術相對的更加精緻細膩，但也使得查找木馬更加困難。DLL hook 技術的使用，往往涉及系統登錄檔記錄的變更，而這也為後續木馬程式的清除作業帶來相當的困擾。

針對前述木馬的特性我們發展出一套偵測及清除木馬程式的機制流程。此機制分成三個部分，分別為：木馬連線確認、木馬程序確認，及木馬程式清除。以下即針對此三部分分別說明。

一、 木馬連線確認

為了克服木馬程式隱藏連線的問題，我們在本文中提出”系統連線特徵值”的概念。何謂連線特徵值？系統在運行一段時間後，其網路往往會多次重覆地與特定遠端位址建立連線；倘若這些連線位址是可受信任的，我們即稱這些位址是系統的連線特徵值。尋找系統內可能的木馬程式連線的過程，實際上等同於求取連線特徵值的相反操作。以下我們即說明求取連線特徵值的方式，並藉此推導出尋找疑似木馬程式連線的準則。

（一） 記錄一長期連線使用狀況資訊

一般系統中所執行的應用程式其使用往往具有週期性，因此對應於特定程式的網路連線也常會週期性的出現。除此之外，隱藏機制較為完全的木馬在程式中往往設有計時器，不在計時器設定的時間根本不會與遠端建立連線。傳統掃描系統通訊埠使用狀況的工具程式由於僅能檢視某一時間點的連線狀況，對於瞬

間出現且連線時間極短的程式常常無法掌握。考量這些情況，選定一較長時間間隔（例如一週）來記錄系統連線使用狀況，是求取連線特徵值的首要步驟。

（二） 判定連線是否可受信任

當步驟（一）完成後將會產出一串 IP 位址清單。在清單上的連線位址是系統連線特徵值與可疑連線的集合。在此階段，我們必須一一檢視存在於集合中的位址，並嘗試判定這些位址 IP 是否分屬可受信任的所有者所擁有。在判定過程中除了經驗法則的運用外，事實上可以藉由一些方式，來協助判斷工作的進行。

1. 透過私用 IP 確認

一般採用傳統 IPv4 建置內部區域網路架構的企業為了減少實體 IP 的使用，通常會借助私有 IP 的採用來配置系統網路位址。基於此種情形，企業內部網路封包收送的兩方其位址必然以私有 IP 的方式呈現。由於一般木馬程式的連線標的通常是實體 IP，因此根據上述觀察結果，當企業內部系統的連線記錄出現私有 IP 格式的位址時，我們通常可以將其納入系統的連線特徵值清單中。採取這樣的判斷基準可以有效的減少待確認的連線數目。

2. 透過 whois 來辨識對外連線

對於不存在連線特徵值清單中的連線 IP，基本上被歸類為”信賴度受質疑”的連線。這些連線或許全為合法連線，只是未能辨識劃歸於連線特徵值中，但更有可能的是，這些連線是由木馬程式所建立。要判斷連線特徵值外的連線是否可疑，往往需要相當的實務經驗，因此難以藉由程式自動化篩選的方式來達成。然而儘管如此，透過 whois 查詢系統的協助，我們可以確實降低辨識連線位址是否值得信賴的困難度。

藉由 whois 查詢，我們可以收集到連線 IP 擁有者的相關資料。透過這些資料，我們可以檢查本地系統是否應該與遠端 IP 存在連線來判斷該程式的合理性。舉例來說，若某一連線的遠端 IP 歸屬於一個本地系統無論如何都不應存在連線的對象，那麼我們通常可以判定系統中存在木馬程式，系統被木馬植入的可能性應該不低。一旦我們排除掉可能的木馬連線之後，剩下的連線 IP 即能將其加入連線特徵值清單中供反覆使用。若能建立資料庫儲存連線特徵值，未來檢查其他系統木馬是否存在時也能得利於此。

3. 減少連線紀錄數目之產出以加快辨識連線特徵值之程序

對於不存在企業內部區域網路的系統，其網路連線少了私有 IP 的存在，因此在整體上會呈現出較企業內部系統之網路連線更加多元的情況。以家庭用個人電腦為例，雖然基於使用者的習慣性，某些網路連線 IP 重複出現的頻率會較高，但大抵上也經常存在許多難以立刻辨識出來的 IP 位址。我們曾經提及，要確認連線是否可受信任是一件不容易的事，因此若能減少待確認的遠端連線 IP，即能加快對於系統連線特徵值的建立。

二、 木馬程序確認

一旦經由系統連線特徵值的比對發現可疑連線，搜尋系統中木馬程式的工作即進入本階段。此階段中透過對系統處理程序與服務的搜索，我們可以據此推斷隱藏著的木馬程式本體何在。

儘管各類木馬程式匿蹤型態各異，但是只要能偵測到木馬執行時所建立的異常連線，就能減少尋找木馬本體的困難度。有鑑於此，我們才會在前一小節中重覆強調系統連線特徵值的重要性。一般在進行連線記錄時，通常會將與連線對應的執行程式名稱與 PID 一併登載。而當連線被劃入可疑連線的集合後，我們即可就記錄的 PID 檢查連線對應的執行程序，來確認木馬是否隱藏於其中。

對於舊式木馬程式而言，由於本身即採取獨立程序的方式執行，因此只要獲悉其 PID，我們就能輕易的找出木馬程式本體所在。不過現今木馬為了更有效的隱藏自身，早已揚棄上述獨立程序執行的運作方式，取而代之的，是採用服務、替換系統程式或是使用 DLL hook 技術進行匿蹤的木馬。

針對以這些型態方式執行的木馬程式，我們通常必須借助如 process explorer 等程序監測工具（圖 1）的協助，來進行搜尋系統中可疑木馬的存在。以下即針對尋找上述不同隱藏執行型態之木馬的方式進行說明。

1. 當可疑連線來自系統程序或是源自一看似合法之應用軟體程序

此時，我們必須考慮下列可能性：

(1) 相關程式是否遭木馬置換或注入：一般來說，要判定程式是否正常，通常可以藉由程式 MD5 的檢查來達成。

(2) 木馬以 rootkit 的方式在程式背後運作：這種情況目前在實務上日益常見。此類木馬除了極為隱蔽而不易搜尋外，要確認其木馬身分也必須具有相當經驗。實務上我們往往須將執行中程式所引用的動態函式庫（DLL）檔案逐一挑出，並搭配網際網路搜尋引擎查明每一個 DLL 的實際用途，方能認定木馬是否存在於其中。

2. 當可疑連線來自於系統服務

當木馬程式隱藏於系統服務內執行時，其搜尋方式如同 1.(2)。然而值得注意的是，在實務上我們發現以 1.(2)方式執行之木馬，大多會伴隨一個到數個不等以服務執行的分身，以便於清除者於清除木馬疏忽之際仍能發揮效用繼續執行木馬工作。

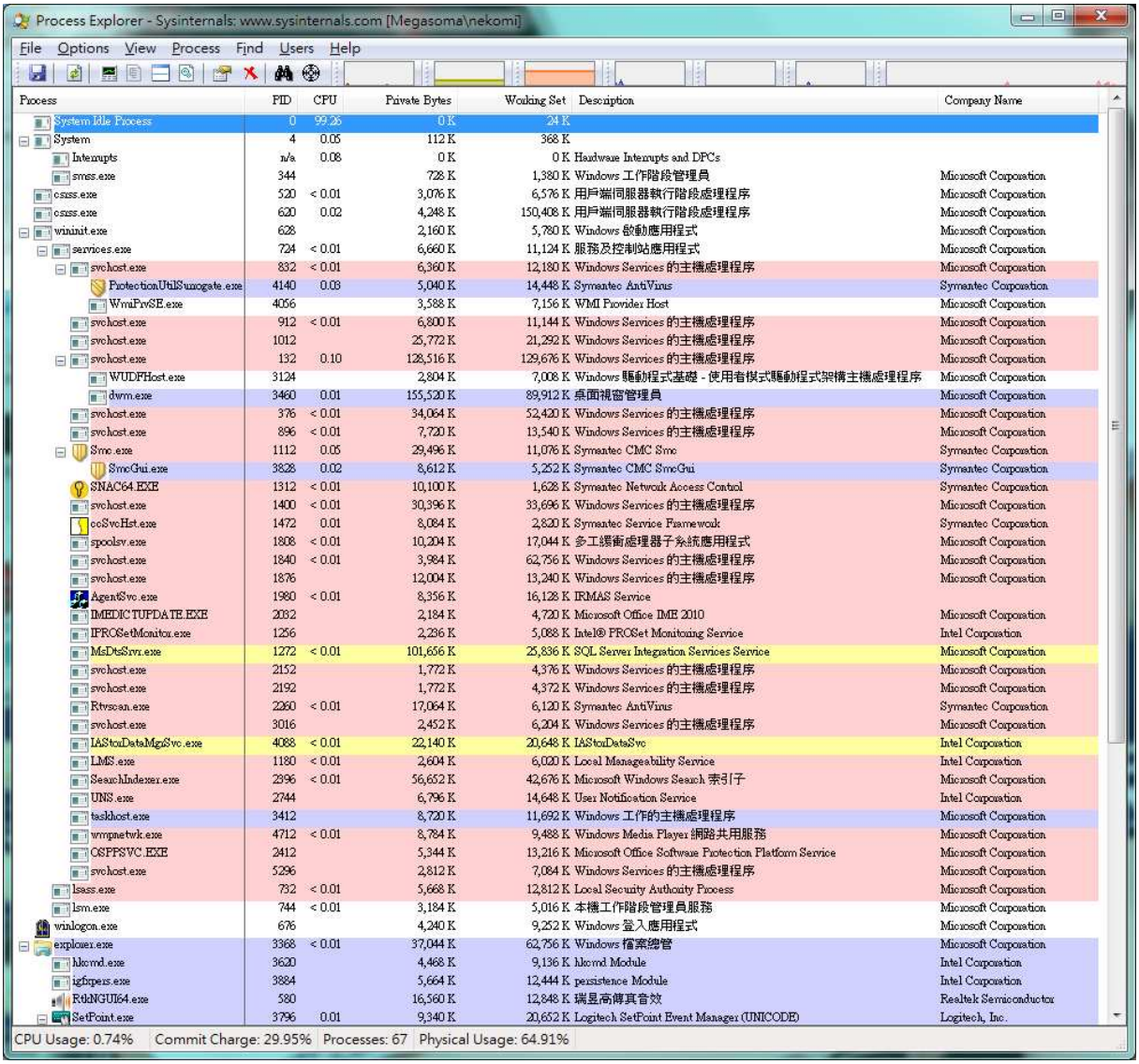

圖 1 Process Explorer 軟體介面

三、 木馬程式清除

一旦確認木馬程式在系統中確實存在且執行中，最後的工作即是清除木馬並使系統恢復未受木馬入侵前的狀態。在 Windows 系統中清除木馬程式必須考慮兩個層面，一個層面在於清除實質存在於系統目錄中的木馬程式本體，另一方面則必須抹除註冊於登錄檔 Registry 中的木馬程式設定。上述的清除工作若能善用系統及登錄編輯程式的搜尋功能，即能達成相當的成果。

實務上對於清除木馬程式尚有是否必須重啟系統的考量。一般而言，為了清除殘存系統記憶體內仍運作中的木馬程序，在進行木馬清除工作後最好重啟系統。但是對於某些要求高可用率的系統往往會要求盡量減少重啟系統的次數，針對此種狀況，我們建議正式清除木馬程式前先執行 kill 指令，移除存在記憶體運作中的木馬，如此可省卻清除木馬程式後重新開機的步驟。

綜合上述分析，我們將偵測清除 Windows 系統中木馬程式的機制以圖形的方式表現如下：

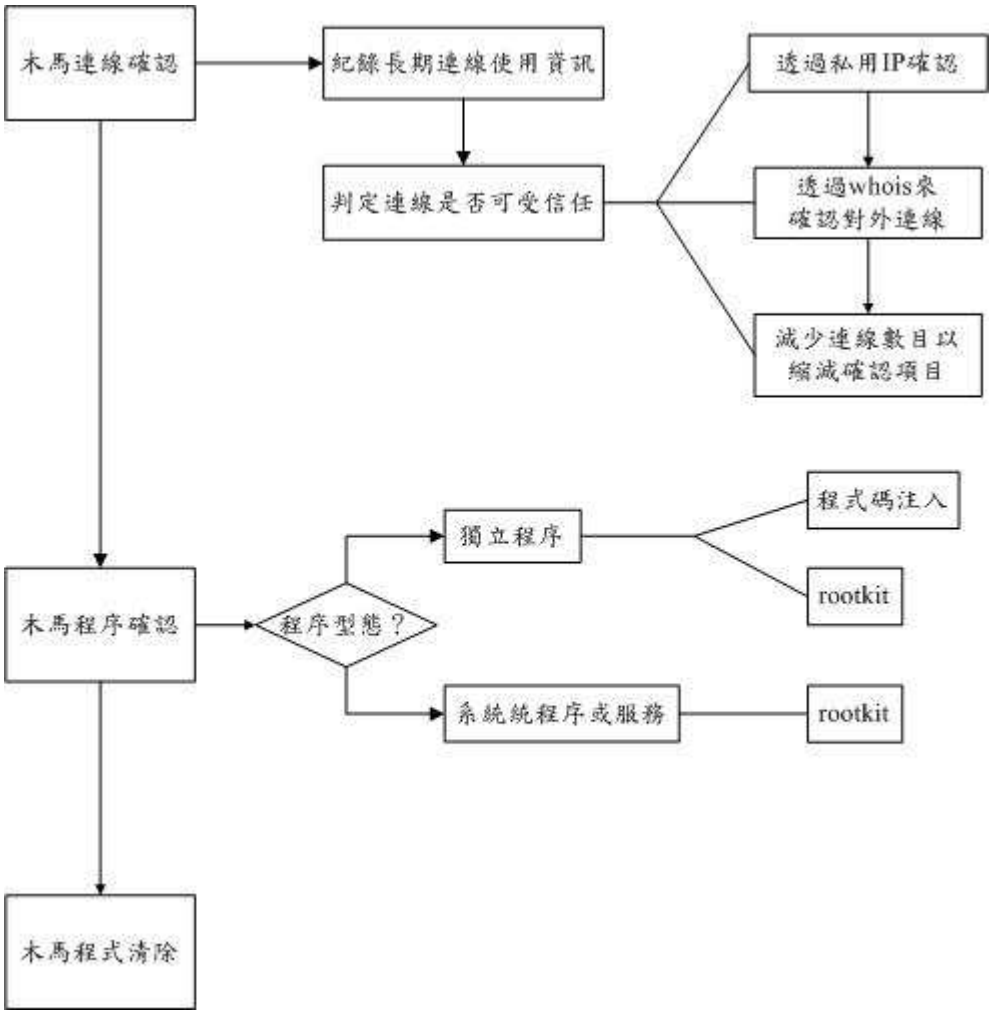

圖 2 偵測清除 Windows 系統木馬程式機制

參、結論與建議

截至目前為止，透過上述的木馬偵測清除機制，我們已經成功的處理相當多系統隱藏木馬的案例。這些案例中不乏系統安置於防火牆與防毒軟體的周全保護下者。總結這些實務經驗來看，此木馬偵測清除機制應具有相當程度的可靠性。而目前我們亦持續對此機制進行改善，冀望於未來能達成下列目標：(1)針對網路連線特徵值進行更深入的研究，(2)根據文中提出的機制，完成木馬偵測清除自動化的作業。除此之外，亦希望藉由此文提供系統管理者一個有別於傳統資訊安全防護之外的參考。

參考文獻

1. 阮寧君，2007，『端口反彈型木馬通信技術研究及防範措施』，信息安全與通信保密，第 12 期。
2. 林小進、錢江，2007，『特洛伊木馬隱藏技術研究』，微計算機信息，第 23 卷・第 33 期。
3. 梁曉、李毅超、崔甲、曹躍，2007，『基於系統調用掛鉤的隱蔽木馬程序檢測方法』，計算機工程，第 33 卷・第 20 期。
4. Dave Kleiman, The Official CHFI Study Guide (Exam 312-49) for Computer Hacking Forensic Investigators, Syngress Publishing, Inc., Burlington, 2007
5. Jie Qin, Huijuan Yan, Qun Si, Fuliang Yan, “A Trojan horse Detection Technology Based on Behavior Analysis”. Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on 23-25 Sept. 2010, pp. 1-4
6. Shugang Tang, “The Detection of Trojan Horse Based on Data Mining”. Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on 14-16 Aug. 2009, pp. 311-314
7. Naiqi Wu, Yanming Qian, and Guiqing Chen, “A Novel Approach to Trojan Horse Detection by Process Tracing” Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference, pp. 721-726

A Trojan horse Detection and Removing Mechanism designed for Windows System

Wei-Jen Chang-Chien
Telecommunication Lab., Chunghwa Telecom Co. Ltd.
williamc@cht.com.tw

Abstract

With the evolution of hackers' skills, the information security protection systems in nowadays are getting more and more difficult to prevent invasions and detect stealth of Trojan horses. To avoid their hiding in computer systems, it is urgent to develop a feasible and effective way to detect and remove these malicious programs. In this article, we constructed a mechanism to detect and remove Trojan horses in Windows system by analyzing behaviors of Trojan horses and integrating practice experiences of production sites. With operations of this mechanism in practical environment, we can confirm the reliability of it. It is also our hope to share this discovering to those system administrators who work hard to reject Trojan horses outside their systems.

Keywords: Trojan horse detection, Trojan horse behavior